AFRL-VA-WP-TP-2006-324

SMOOTH FUNCTION MODELING FOR ON-LINE TRAJECTORY RESHAPING APPLICATION (POSTPRINT)

Ajay Verma, Kalyan Vadakkeveedu, Michael W. Oppenheimer, and David B. Doman

JULY 2006

THIS IS A SMALL BUSINESS INNOVATION RESEARCH (SBIR) PHASE II REPORT.

Approved for public release; distribution is unlimited.

STINFO COPY

AIR VEHICLES DIRECTORATE
AIR FORCE MATERIEL COMMAND
AIR FORCE RESEARCH LABORATORY
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7542

NOTICE AND SIGNATURE PAGE

AFRL-VA-WP-TP-2006-324 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

*//Signature//

Michael W. Oppenheimer
Electronics Engineer
Control Design and Analysis Branch
Air Force Research Laboratory
Air Vehicles Directorate

//Signature//

Deborah S. Grismer
Chief
Control Design and Analysis Branch
Air Force Research Laboratory
Air Vehicles Directorate

//Signature//

Jeffrey C. Tromp
Senior Technical Advisor
Control Sciences Division
Air Vehicles Directorate

*Disseminated copies will show "//Signature//" stamped or typed above the signature blocks.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE ***(DD-MM-YY)***	**2. REPORT TYPE**	**3. DATES COVERED** ***(From - To)***
July 2006	Conference Paper Preprint	06/01/2005– 06/07/2006

4. TITLE AND SUBTITLE
SMOOTH FUNCTION MODELING FOR ON-LINE TRAJECTORY RESHAPING APPLICATION (POSTPRINT)

5a. CONTRACT NUMBER FA8650-05-C-3505
5b. GRANT NUMBER
5c. PROGRAM ELEMENT NUMBER 0605502

6. AUTHOR(S)
Ajay Verma and Kalyan Vadakkeveedu (Knowledge Based Systems, Inc.)
Michael W. Oppenheimer and David B. Doman (AFRL/VACA)

5d. PROJECT NUMBER A05G
5e. TASK NUMBER
5f. WORK UNIT NUMBER 0C

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Knowledge Based Systems, Inc. 1408 University Dr. College Station, TX 77840	Control Design and Analysis Branch (AFRL/VACA) Control Sciences Division Air Vehicles Directorate Air Force Materiel Command, Air Force Research Laboratory Wright-Patterson Air Force Base, OH 45433-7542

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Vehicles Directorate
Air Force Research Laboratory
Air Force Materiel Command
Wright-Patterson Air Force Base, OH 45433-7542

10. SPONSORING/MONITORING AGENCY ACRONYM(S) AFRL-VA-WP

11. SPONSORING/MONITORING AGENCY REPORT NUMBER(S) AFRL-VA-WP-TP-2006-324

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES
This work has been submitted to Proceedings of the 2006 AIAA Guidance Navigation, and Control Conference, published by AIAA.
This is a Small Business Innovation Research (SBIR) Phase II Report. PAO Case Number: AFRL/WS 06-1833 (cleared July 27, 2006).

14. ABSTRACT
This report was developed under a SBIR contract.
Online vehicle trajectory reshaping is desired for a class of autonomous air vehicles such as RLVs in order to avoid catastrophic failure when subjected to performance restricting damages and failures. An *Adaptive Trajectory Reshaping and Control* (ATRC) system is envisioned that responds to altered vehicle conditions by continuously retargeting and reshaping the reference RLV trajectory satisfying the feasibility constraints. On-line trajectory reshaping to determine a feasible reference trajectory is computationally a difficult problem for real time applications. ATRC is exploring the principles of vehicle dynamics inversion for online generation of feasible reference trajectory. Two essential components for generating reference trajectory for air-vehicles using "inverse dynamics" methodology are aerodynamic model of the vehicle that is representative of the current state of the vehicle, and a framework for modeling the vehicle trajectory. Physics based modeling software such as Missile DATCOM allows fast computation of aerodynamic coefficients for given flight points and the results can be stored in tabular form. However, for efficient real-time trajectory reshaping application, it is desirable to represent aerodynamic coefficients in smooth functional forms that are governed by a few parameters. Similarly, trajectories must also be represented by smooth functions. In this paper we present modeling of smooth functions using a set of basis functions that are suitable for aerodynamic modeling and trajectory reshaping of the air vehicles. A desirable feature for function modeling is the easy imposition of boundary as well as mid point constraints in the function using a small number of parameters without limiting the scope of the function. In this paper we present a design method for generating orthonormal polynomial basis functions in one and two dimensions with constraints.

15. SUBJECT TERMS SBIR Report, Trajectory Reshaping, inverse dynamics, aerodynamic model

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	**17. LIMITATION OF ABSTRACT:**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** (Monitor)
Unclassified	Unclassified	Unclassified	SAR	22	Michael W. Oppenheimer **19b. TELEPHONE NUMBER** *(Include Area Code)* N/A

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18

EXTENDED ABSTRACT

Smooth Function Modeling for On-Line Trajectory Reshaping Application

Ajay Verma*, Kalyan Vadakkeveedu†
Knowledge Based Systems, Inc.

Michael W. Oppenheimer‡ and David B. Doman§
AFRL/VACA

Online vehicle trajectory reshaping is desired for a class of autonomous air vehicles such as RLVs in order to avoid catastrophic failure when subjected to performance restricting damages and failures. An *Adaptive Trajectory Reshaping and Control*[1] (ATRC) system is envisioned that responds to altered vehicle conditions by continuously retargeting and reshaping the reference RLV trajectory satisfying the feasibility constraints. On-line trajectory reshaping to determine a feasible reference trajectory is computationally a difficult problem for real time applications. ATRC is exploring the principles of vehicle dynamics inversion for on-line generation of feasible reference trajectory. Two essential components for generating reference trajectory for air-vehicles using "inverse dynamics" methodology are aerodynamic model of the vehicle that is representative of the current state of the vehicle, and a framework for modeling the vehicle trajectory. Physics based modeling software such as Missile DATCOM allows fast computation of aerodynamic coefficients for given flight points and the results can be stored in tabular form. However, for efficient real-time trajectory reshaping application, it is desirable to represent aerodynamic coefficients in smooth functional forms that are governed by a few parameters. Similarly, trajectories must also be represented by smooth functions. In this paper we present modeling of smooth functions using a set of basis functions that are suitable for aerodynamic modeling and trajectory reshaping of the air vehicles. A desirable feature for function modeling is the easy imposition of boundary as well as mid point constraints in the function using a small number of parameters without limiting the scope of the function. In this paper we present a design method for generating orthonormal polynomial basis functions in one and two dimensions with constraints.

I. Introduction

The large potential for space utilization is not being exploited as it is currently inhibited by the huge cost of launching operations. The benefits of advanced space utilization can be greatly increased by making space utilization more affordable The Reusable Launch Vehicle (RLV) programs are targeted towards affordable space utilization. However, to maintain the economical viability of RLVs, it is important to enhance operational safety and reliability by providing the RLV the capability to respond to various uncertainties and evolving emergency situations. Responding to an uncertain environment after a damage/failure presents many tough technical problems for this class of vehicles. These problem manifests

* Senior Researcher, 1408 University Dr, College Station, TX-77840, Senior Member AIAA, averma@KBSI.com.
†Researcher, 1408 University Dr, College Station, TX-77840, Member AIAA, kvadakkeveedu@KBSI.com.
‡Electronics Engineer, 2210 Eighth Street, Bldg. 146, Rm. 305, WPAFB, OH 45433-7531, Member AIAA, Member AIAA, Michael.Oppenheimer@wpafb.af.mil
§Senior Aerospace Engineer, 2210 Eighth Street, Bldg. 146, Rm. 305, WPAFB, OH 45433-7531, Associate Fellow AIAA, David.Doman@wpafb.af.mil

in the following challenges that must be overcome: First, to adequately determine and model the dynamic characteristics of the vehicle in the altered state after a damage/failure; second, to estimate the new constraints and limitation(s) of the vehicle; third, to adapt and reconfigure the command, control and guidance of the vehicle to the modified system dynamics; and fourth, to design and plan a new feasible path with respect to the end goal maximization. A high percentage of such damage/failure cases leave the vehicle in an uncontrolled and uncertain environment with a high probability of ultimately entering a state of catastrophic failure. The high cost of loss resulting from catastrophic failures, has prompted researchers in the direction of developing technologies to assist in minimizing such failures.

Damage to a vehicle or a sub-system failure may result in modification of the applicable trajectory constraints and/or the dynamical behavior of the system, consequently making the previously designed reference trajectory infeasible. An acceptable trajectory for a dynamical system is a solution of a two-point boundary value problem for a set of governing differential equation of motion. The real world systems such as air vehicles are highly non-linear systems and impose a set of constraints on the trajectory variables as well as control variables. In inverse dynamics approach, a trajectory is specified first, which results in solving a set of algebraic equations, yet strictly satisfying the non-linear differential equations of a non-flat system.

In this paper first we introduce the architecture of an Adaptive Trajectory Reshaping and Control (ATRC) system for the general class of RLV systems, which is based on the principles as described in Ref. [1]. Next we discuss the general inverse dynamics approach for trajectory reshaping of RLVs and the motivation for functional modeling. For trajectory determination, there are two types of functions that must be modeled. The first set of functions is needed to describe the vehicle trajectory. The functions for the spatial coordinates of the vehicle constitute a vehicle trajectory. These trajectory functions are normally functions of one independent variable. The second set of functions that must be modeled in real time is the modified aerodynamic constraints. The aerodynamic constraints are modeled by defining aerodynamic coefficients for the flight envelope of the vehicle. Mostly, these functions are two dimensional with *Mach* and *angle of attack* being the independent parameters. For faster convergence, it is desired that the functions be smooth, with continuity in the value as well as first derivative. A general approach to model a function is to parameterize the function and then determine the parameters that satisfy any governing constraints to a satisfactory level. In Section IV, we describe a novel approach for designing a set of basis functions with a class of constraints built in, that reduces the complexity of the solution and results in better approximations.

II. Adaptive Trajectory Reshaping and Control (ATRC) System

The ATRC system enhances RLV capability to avoid catastrophic failure when subjected to performance restricting damages and failures. The overall goal of ATRC translates into specific requirements for design and development of functionalities related to adaptable and reconfigurable command, control, and guidance system for the RLVs. and real time solution techniques for generating feasible trajectories.

Figure 1 shows the general architecture of the envisioned ATRC system for RLVs. Note that the above structure is specific to longitudinal motion of the vehicle; however, it can be easily extended to the six degrees of freedom. The main components of the envisioned ATRC system requires:

1. On-line system identification that includes physics based modeling that accounts for vehicle damage, parameter estimation and parameter projection for constraint boundary determination. The constraint boundaries influence the trajectory reshaping of the vehicle.
2. Real time trajectory determination for reshaping reference trajectory under feasibility constraints.
3. Adaptive, closed loop control and guidance system for reference trajectory tracking.

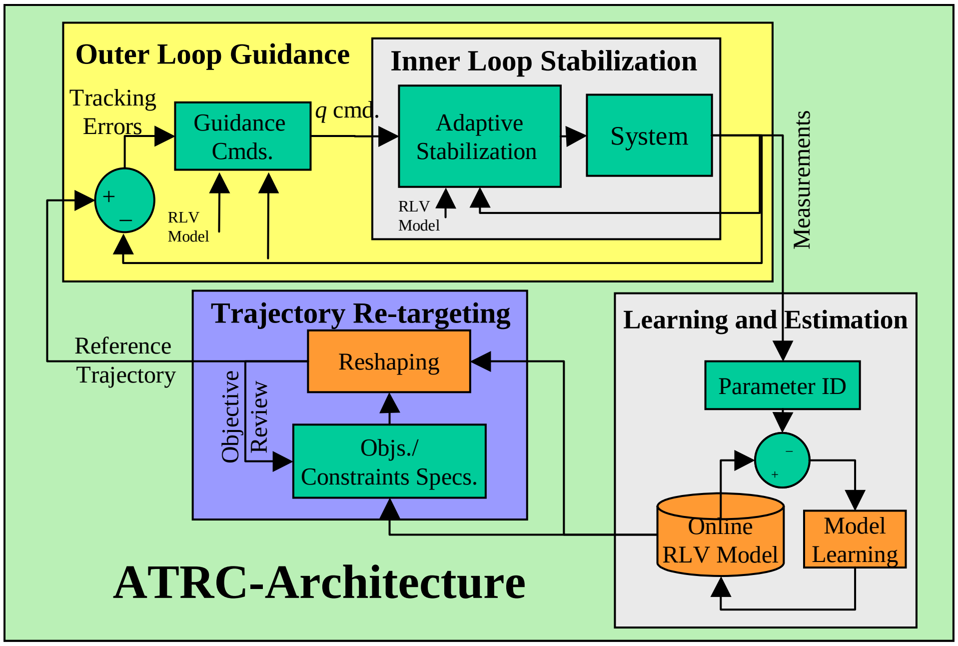


Figure 1. Architecture of ATRC

III. Inverse Dynamics Approach for Trajectory Generation

The advantage of inverse dynamics approach is that one can avoid integration of differential equations of motion altogether. The main difference between regular trajectory generation approach and inverse dynamics approach is that that in the latter approach we first parameterize the trajectory and then use numerical techniques to solve for these parameters that minimize the objective function and satisfy other inequality constraints. Note that once a smooth trajectory is specified, the time derivatives of the trajectory parameters are also fixed. With the availability of state and its time derivatives, the differential equations become simple algebraic equations, which is much faster to solve than differential equations.

A. Reference Trajectory Design

To determine a trajectory for a non-linear dynamic system, a solution must be found that satisfies the set of differential equations governing the dynamics of the system. Further, the trajectory solution should not violate some non-linear constraints, which limits the operation capability of the system. For an aircraft, the constraints arise due to several factors such as limitations on the *angle of attack*, *load factor*, *aerodynamic heating*, and *actuator saturation*.

There are two primary approaches for trajectory generation and these have been classified in the literature [2] as the "integral approach" and the "differential approach." In any approach, where generation of a trajectory involves the integration of the equations of motion, this approach is classified as the "integral approach." In a differential approach, an assumed functional form for trajectory is differentiated to obtain algebraic functions for the higher derivatives, which are required to impose constraints on the control inputs for the "inverse dynamics" solution. There are various applications where inverse dynamics have been used, such as spacecraft trajectories and path planning in robotics [3] and overhead cranes [4]. Historically, the inverse dynamics approach has been used for "differentially flat" systems. A system is "differentially flat" [see 5,6] if there exists a set of outputs, known as "flat outputs," such that there is a one-to-one correspondence between the trajectories of flat outputs and the full state and control inputs of the system. In our approach, we use an inverse dynamics approach for determining trajectory as this allows us to solve algebraic equations instead of integrating ODEs. With the inverse dynamics approach for aircraft trajectories, a problem arises due to inherent under-actuation in most of the aircrafts. For a six-degrees-of-freedom (6DOF) aircraft, there are normally four controls: thrust, elevator, aileron, and rudder. Ref. [7] defines a novel trajectory generation scheme, which uses pseudo forces for inverse dynamic computation. In this paper we concentrate on function modeling approach to make the inverse dynamic solution approach more efficient.

B. Inverse Dynamics Approach

Assume x be the state vector for a vehicle and $x = x(\tau)$ represent a reference trajectory of the vehicle. The goal of the inverse dynamic approach is to design an inverse trajectory such that the trajectory and the corresponding solution for control inputs satisfy all constraints. The governing equations of motion for an air vehicle may be approximated with terms that are non-linear with respect to some of the states and linear with respect to some others. For example, if x_1 and x_2 are the subset of the state vector x, a typical governing equation may be written in the form

$$\dot{x} = C_1(x_1) + C_2(x_1)x_2 + C_3(x_1)u \ , \tag{1}$$

where $C_1(x_x), C_2(x_x), C_3(x_x)$ are non-linear functions in x_1, and the over all function is linear with respect to x_2 and control inputs u. For example, the pitch dynamics of a vehicle for longitudinal dynamics can be written as

$$I_{YY}\dot{q} = C_{m0}(M,\alpha) + C_{m_q}(M,\alpha)q + C_{m_{\delta e}}(M,\alpha)\delta e \ , \tag{2}$$

where $x_1 = [M,\alpha]$, $u = \delta e$, and C_1, C_2, C_3 are $C_{m0}/I_{YY}, C_{m_q}/I_{YY}, C_{m_{\delta e}}/I_{YY}$ respectively. The inverse dynamic solution for Eq. (1) can be written as

$$u = C_3^{-1}(x_1)(\dot{x} - C_1(x_1) + C_2(x_1)x_2) \tag{3}$$

However, note that the existence of C_3^{-1} may not always be guaranteed for any arbitrary trajectory. In the event of any damage to the RLV , the reference trajectory must be re-designed in view of the altered vehicle dynamics. For a feasible solution, a numerical iterative approach is used [7], where the reference trajectory is perturbed until feasibility is ensured. To facilitate perturbation of the vehicle trajectory, the trajectory must be parameterized. Generally, the trajectory design must meet certain constraints on end points, and sometimes there may be constraint at mid points. In this paper we will present functional modeling that allows function parameterization as well as ensures any constraints on the reference trajectory. For fast convergence of the numerical solution to the inverse problem , it is desired that the vehicle model, represented by terms C_i, $i = 1,\cdots,3$ be smooth functions. As the vehicle dynamics are altered, the altered coefficient functions $C_i(x_1)$ must be re-determined on-line.

If the damage condition is known, physics based modeling techniques, such as the approach used by Missile DATCOM [8], can determine the numerical values of the coefficient functions at various flight points. In our approach, we first determine the vehicle coefficients in tabular form spanning the designed flight envelope. However, we require a framework to capture the coefficients in functional forms such that the non-linear functions are continuous and smooth. In the following sections we address the design of a set of constrained orthonormal bases, followed by one dimensional trajectory modeling, and two-dimensional coefficient functional representation in parametric form.

IV. Constrained Orthonormal Polynomial Basis Functions

In this section we present an approach to design orthonormal polynomial basis functions with desired constraints built in. The choice of basis functions ultimately influences the complexity of modeling the desired functions or behaviors. It is well known that a sub set of a complete set of basis functions can approximate any given function to a desired accuracy by choosing a sufficient number of basis functions elements in the sub set. However, if a function to be approximated must meet certain constraints, it presents various problems. First, a finite set of basis function may not ensure the constraint- satisfaction on the function to be approximated. Second, a large number of basis functions may be required for the satisfactory constrained approximation. Now, if we can create a set of basis functions, where all the basis functions satisfy the given constraints, any linear combination of those functions will also satisfy the given constraints. With this motivation we developed an approach to design a constrained orthonormal polynomial basis functions. To explain our approach, we first present a way for generating one-dimensional unconstrained orthonormal polynomial based basis functions and then extend it to generate constrained orthonormal polynomial basis functions.

A. One Dimensional Basis Functions with Built-In Constraints

Let $\Phi_i(x)$ represent the i^{th} 1D basis function of the orthonormal polynomial basis set given as:

$$\Phi_i(x) = \sum_{k=0,i} a_k x^k, \quad i = 0,1,\cdots \tag{4}$$

Notice that the i^{th} basis function has i+1 polynomial coefficients to be determined. We impose the given constraints on each of the basis functions. E.g. if a desired constraint is that the modeled function g(x) should be zero at x = 0.5, then we create a new basis function set where $\Phi_{C,i}(x) = (\Phi_i(x) \otimes (x-0.5))$, where $\otimes$ is the polynomial multiplication(or 1D convolution of the polynomial coefficients). $g(x)=0$ for all the functions in the space spanned by this new basis set. The Gram-Schmidt procedure is applied to this constrained basis set to generate an orthonormal basis set.
The inner product between any two functions in this space is defined as

$$\langle \Phi_i(x), \Phi_k(x) \rangle = \int_0^1 \Phi_i(x)\Phi_k(x)dx\,. \tag{5}$$

The norm and the orthogonality conditions can be obtained as:

$$\int_0^1 (\Phi_i(x))^2 dx = 1, \quad \text{Normalizing condition.} \tag{6}$$

$$\int_0^1 \Phi_i(x)\Phi_k(x)dx = 0,\ k = 0,1,\cdots,i-1, \quad \text{Orthogonality condition.} \tag{7}$$

The orthogonality condition gives i independent equations while an additional equation from normalization ensures that all the coefficients $a_k, k = 0,i$ can be obtained.

For automatically generating the orthogonal basis set we use the Gram-Schmidt recursive approach. Assuming that we already know $\Phi_k, k = 0, i-1$. Define an arbitrary starting polynomial $f_i^0(x)$ of degree i. For k = 0 to i – 1 we determine $f_i^{k+1}(x)$ as

$$f_i^{k+1}(x) = f_i^k(x) - \langle \Phi_k(x) f_i^k(x) \rangle \Phi_k(x) \tag{8}$$

The i^{th} basis function $\Phi_i(x)$ is now given as

$$\Phi_i(x) = f_i^i(x) \tag{9}$$

Next we will extend the above formulations to impose a given set of constraints. The constraint can be specified on the function value or higher derivatives at specified points. A typical n^{th} order constraints is given as:

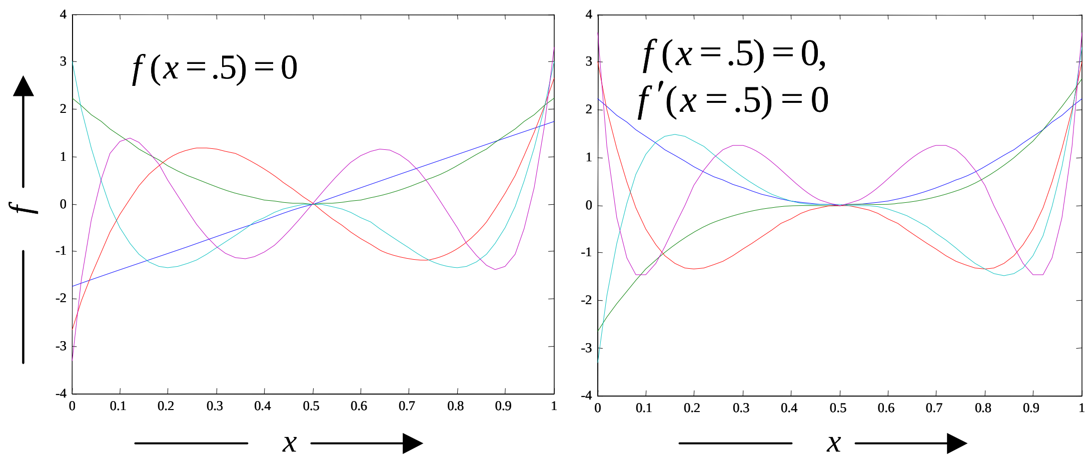
$$\left.\frac{\partial^n \Phi_i(x)}{\partial x^n}\right|_{(x=\alpha)} = 0 \tag{10}$$

Before we describe an approach to determine constrained orthogonal polynomial basis functions, we note some properties of polynomials. A polynomial satisfies an n^{th} order constraint as given in Eq. (10) at $x=\alpha$ when α is the root of the reduced polynomial obtained through n differentiations. Also note that a polynomial of degree m trivially satisfy any constraint of order more than m, while constraints of order m or less must be imposed by adjusting the polynomial coefficients. Sometimes, a polynomial of degree m may not exist that satisfy all the constraints, especially when there are too many constraints of order m and lees that must be imposed. In this case the m^{th} degree polynomial will not be a part of constraint orthogonal basis functions.

For constrained orthogonal polynomial function, first we define a set of polynomial functions of all possible degrees that satisfy the given constraints. Next, we use Gram-Schmidt scheme to orthogonalize and normalize the functions to form a basis function set. The i^{th} function $f_i(x)$ starting from $i=0$ is determined in the following iterative manner.

Set $f_i^0(x)=x^i$.

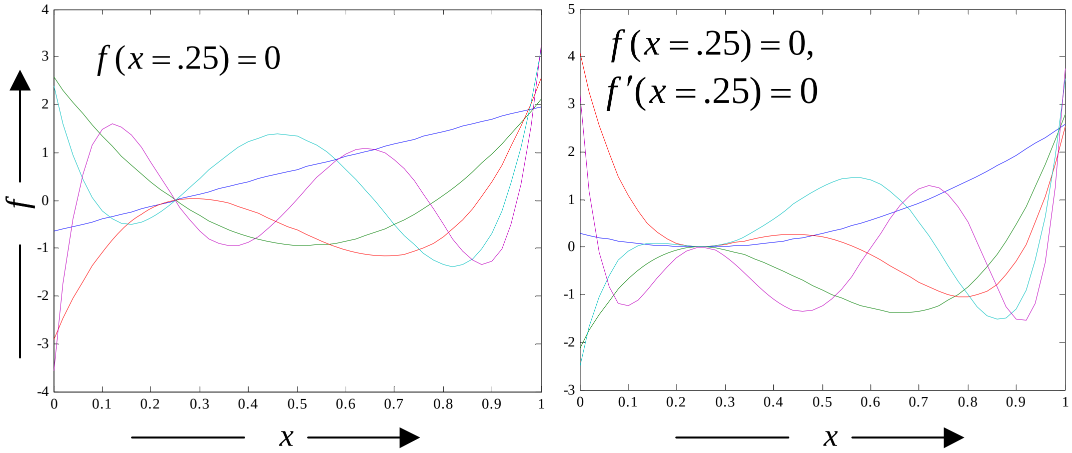
Next extract the lowest order constraint from the given set of constraints that must be imposed. If the lowest order constraint has order *m*, and $m<\text{degree}\left(f_i^k\right)$, then incorporate the constraint. Incorporating a constraint also increases the degree of the polynomial function. Repeat the process for the next lowest order constraint, until the lowest order constraint is equal to the current degree of $\left(f_i^k\right)$. Figure 2 and Figure 3 show some examples of a class of orthogonal basis functions that incorporate the given constraints. The example presents normalized orthogonal basis functions in the range 0 to 1.



(a) **(b)**

Figure 2. Constrained Polynomial Basis Functions. (a) Basis Function are constrained to be Zero at *x* = 0.5. (b) Both, function and its derivative are constrained to be Zeros at *x* = 0.5

In Figure 2(a), the value of the function is constrained to be zero mid way, i.e. $f(x)\big|_{x=0.5}=0$ Figure 2(b) adds an additional constraint on the slope of the curve or the first derivative of the function given as $df(x)/dx\big|_{x=0.5}=0$. In Figure 3, the application of the constraint has been shifted to the quarter point, i.e. $x=0.25$. In this case, the basic nature of orthogonal basis functions remains the same, albeit with a loss of symmetry. Figure 2 and Figure 3 demonstrate the ability to generate a range of readily available sets of *constrained orthogonal polynomial basis functions*.



(a) **(b)**

Figure 3. Constraint Polynomial Basis Functions. (a) Basis Function are Constraint to be Zero at *x* = 0.25. (b) Both, Function and its Derivative are Constrained to be Zeros at *x* = 0.25

B. Two-Dimensions Ortho-Normal Polynomials Basis Functions

In this section we demonstrate an approach to produce a set of polynomial orthonormal basis functions in two dimensions, represented by x and y with maximum degree of N,M respectively. First we introduce the vector space of two-dimensional polynomials and then the algorithm to generate orthonormal basis vectors.

1. *Matrix Representation of 2D Polynomials*

Let $\mathbf{P}_{N,M}$ be a set of polynomials where $P(x,y)\in\mathbf{P}_{N,M}$ is a polynomial in x and y of degree N and M respectively. Let the coefficients be a_{ij} .

$$P(x,y)=a_{NM}x^N y^M + a_{N-1M}x^{N-1}y^M + \cdots + a_{0M}x^0 y^M + a_{N-1M}x^{N-1}y^M + \cdots + a_{00}x^0 y^0 ,$$

or

$$P(x,y)=\sum_{i=1}^{N}\sum_{j=1}^{M}a_{i,j}\cdot x^{N-i}\cdot y^{M-j} . \qquad (11)$$

Let us represent the polynomial coefficients in matrix form as,

$$A=\begin{bmatrix} a_{NM} & a_{N-1M} & a_{N-2M} & \cdots & a_{0M} \\ a_{NM-1} & a_{N-1M-1} & a_{N-2M-1} & \cdots & a_{0M-1} \\ \vdots & \vdots & \ddots & & \vdots \\ a_{N0} & a_{N-10} & a_{N-2,0} & \cdots & a_{00} \end{bmatrix}_{M+1,N+1} . \qquad (12)$$

Then the polynomial $P(x,y)$ can be represented in matrix form as,

$$P(x,y)=X^T AY , \qquad (13)$$

where,

$$X = \begin{bmatrix} x^N \\ x^{N-1} \\ \vdots \\ 1 \end{bmatrix}_{N+1}, \text{ and } \quad Y = \begin{bmatrix} y^M \\ y^{M-1} \\ \vdots \\ 1 \end{bmatrix}_{M+1}. \tag{14}$$

2. Representation of a 2D Polynomial Inner Product Vector Space

Let $\mathbf{P}_{N,M}$ be a set of polynomials in *x* and *y* of degrees up to *N* and *M* respectively. The polynomial set $\mathbf{P}_{N,M}$ defines a polynomial vector space over ***R*** (set of real numbers) with

$$\Theta = \left\{\theta_{i,j} \mid \theta_{i,j} = x^i y^j \right\}_{0 \le i \le N, 0 \le j \le M} \tag{15}$$

as the basis vectors that satisfy the basic properties of associativity, commutativity, existence of identity and inverse, and properties of scalar multiplication. In order to define orthogonality, we must define an *inner product space*. An *inner product space* is a *vector space* with an additional structure of *inner product.* We define the *inner product* on the 2D polynomial vector space as,

$$\left\langle P(x,y), Q(x,y)\right\rangle = \int_0^1\int_0^1 P(x,y) \otimes Q(x,y)\, dx\, dy, \qquad P(x,y), Q(x,y) \in \mathbf{P}_{N,M}. \tag{16}$$

Our goal is to construct an orthonormal basis for the polynomial vector space $\mathbf{P}_{N,M}$. As a first step we compute an orthonormal basis functions set of vectors using the Gram-Schmidt procedure. Then we approximate the function based on its projection on the basis functions.

3. Constrained Orthonormal Basis Set for 2D Functions

Many times we would like to represent a class of functions in terms of basis functions that must satisfy certain constraints. For an efficient representation, we would like to construct a set of basis functions satisfying the given constraints in addition to the orthonormality condition. In this report we restrict our discussion to three types of constraints; constraints on (i) the function *value*, (ii) function *value* and *first derivative* or (iii) only on the first derivative. In the following we describe the generation of 2D constrained basis functions.

(i) If the constraints are imposed only on the value of the function along the curve *C(x,y)* = 0, the polynomial basis set Θ is modified to introduce the constraints and create a new basis set Θ_C. The new basis set is given by,

$$\Theta_C = \left\{\theta_{i,j}(x,y) \otimes C(x,y)\right\}.$$

The symbol $\otimes$ denotes polynomial multiplication (or equivalently discrete convolution of the polynomial coefficients). Any function *g(x,y)* that lies in the space spanned by Θ_C satisfy the constraint that, along the curve *C(x,y)* = 0, *g(x,y)* = 0.

Note that the basis set Θ_C is not yet orthonormal. The constrained orthonormal basis set Φ_C is constructed using the Gram-Schmidt orthogonalization procedure. See Figure 4 for an example of the orthonormal basis functions generated using this procedure. The basis set Θ_C and the orthonormal basis set Φ_C span the same function space.

(ii) If the constraints are imposed on the value as well as the derivatives along a curve *C(x,y)* = 0, a constrained basis set Θ_C is computed from the original basis set Θ as,

$$\Theta_C = \{\theta_{i,j}(x,y) \otimes C(x,y) \otimes C(x,y)\}.$$

The symbol $\otimes$ denotes polynomial multiplication (or equivalently discrete convolution of the polynomial coefficients). Any function *g(x,y)* that lies in the space spanned by Θ_C satisfy the constraint that, along the curve *C(x,y)* = 0, *g(x,y)* = 0 and *g'(x,y)* = 0. The constrained orthonormal basis set Φ_C is constructed using the Gram-Schmidt procedure as discussed in the previous section. Figure 5 shows an example of the orthonormal basis functions generated.

(iii) Constraint only on the derivative of the function. We then modify the given polynomial Θ as,

$$\Theta_C = \{\theta_{0,0}(x,y)\} \cup \{\Theta - \{\theta_{0,0}(x,y)\}\} \otimes C(x,y) \otimes C(x,y).$$

Figure 6 gives an example of orthonormal functions generated with constraints only on the derivatives.

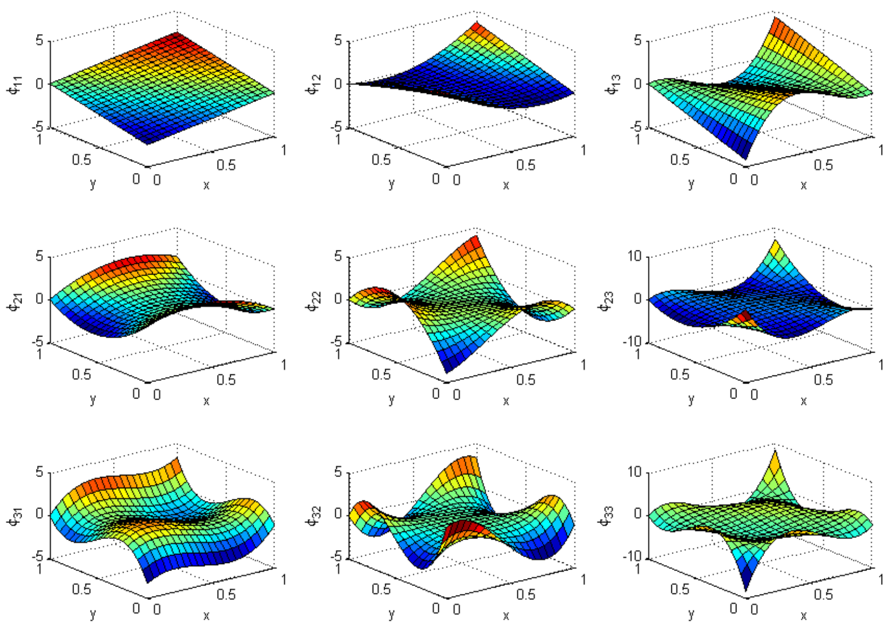


Figure 4. Orthonormal basis function set of polynomials in x and y of degrees 2 and 2, with the constraint that at x+y = 1, the value of the function is zero.

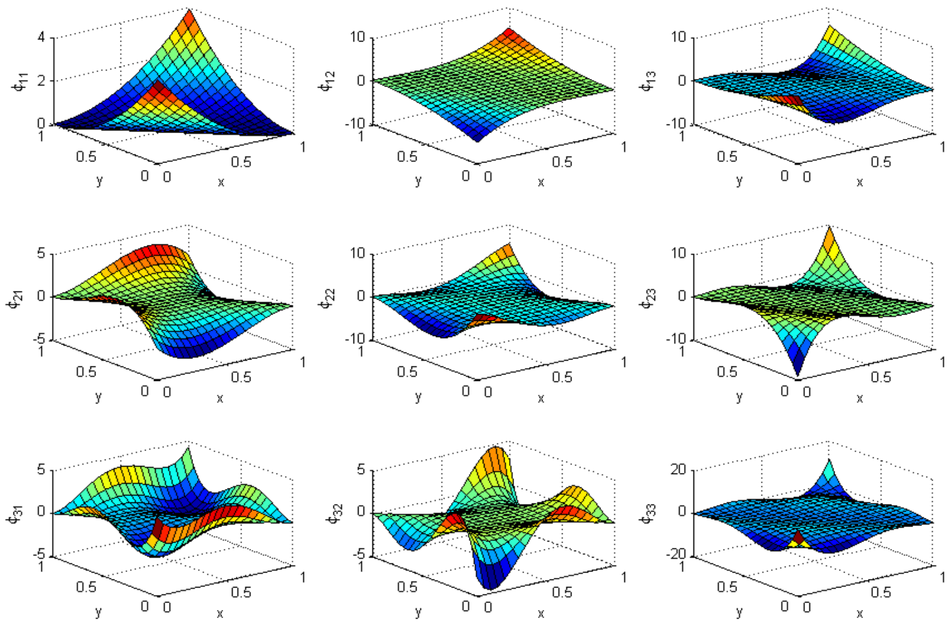


Figure 5. Orthonormal basis function set of polynomials in x and y of degrees 2 and 2, with the constraint that at x+y = 1, the value and derivatives of the function are zero.

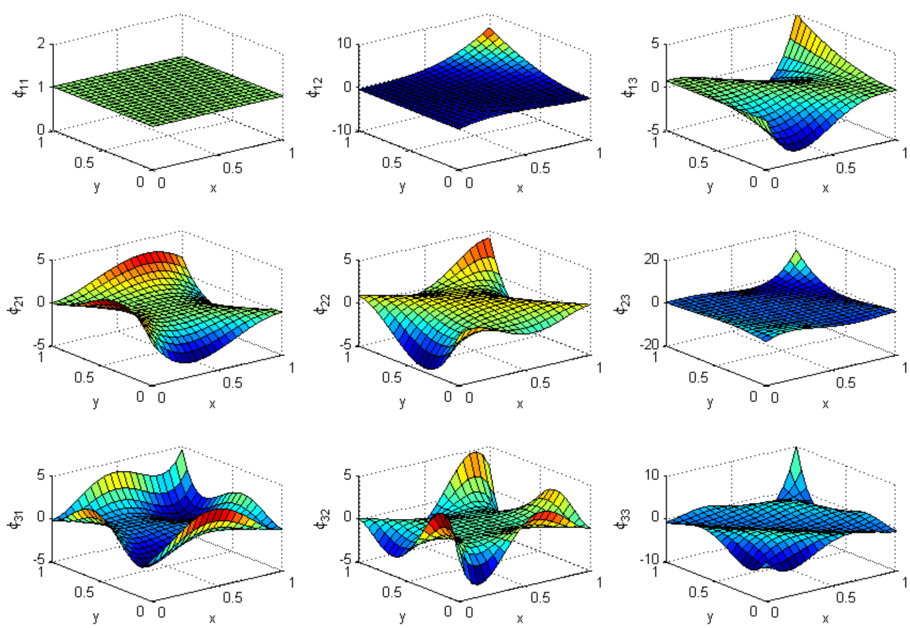


Figure 6. Orthonormal basis function set of polynomials in x and y of degrees 2 and 2, with the constraint that at x+y = 1, the derivative of the function is zero.

V. One-Dimensional Function Modeling with Constraints using Basis Functions

Often we need to estimate or model non-linear functions that must satisfy some constraints. For example, the inverse dynamic solution approach requires parameterization of multiple system output variables or the trajectory variables with some constraints imposed on them. Note that the feasible trajectory must also satisfy the governing differential equations. The trajectory parameters are determined using an optimization process that minimizes the error between trajectory variables and the solution of the governing equations. Note that if some of the trajectory constraints, such as the boundary constraints are forced in the trajectory parameterization itself, it helps in reducing the complexity of the optimization problem. This work extends the approach followed in Verma et al [7,9] for representing trajectory in functional form that is useful to impose boundary and in-point constraints. To prescribe a smooth trajectory functional for a specific trajectory coordinate, the position coordinate is represented by a twice differentiable, smooth function so that velocity and acceleration can be uniquely defined. The path for an individual position coordinate is defined as a function of normalized time $\tau = t / T_f$, where T_f is the total time for the maneuver. An individual coordinate trajectory is structured to have two parts, a *base trajectory* and a *perturbation* of the base trajectory. If $P(\tau)$ is the i^{th} coordinate, it is chosen to be of the form

$$P(\tau) = P_0(\tau) + P_1(\tau) \sum_{j=1}^{l} \alpha_j \varphi_j(\tau) \tag{17}$$

Here $P_0(\tau)$ is the base trajectory function, which is chosen such that it satisfies any boundary or mid point constraints for the trajectory. One example approach $P_0(\tau)$ could be chosen as a minimum degree polynomial spline that satisfies the desired constraint conditions. The base trajectory by itself may not be a true solution of the governing equations and hence an infeasible trajectory. To make the trajectory feasible, we add a second term on the right hand side of the trajectory equation. The perturbation term must be designed to ensure that the overall trajectory function $P(\tau)$ is the solution of the governing equations. Perturbation term is defined using a set of orthonormal basis functions $\{\varphi_j(\tau)\}, j = 1, \cdots, l$ that are modified by the term $P_1(\tau)$. The term $P_1(\tau)$ is a weight polynomial constraining the perturbation term to contribute zero value for the already satisfied desired constraints by base trajectory $P_0(\tau)$. For example $P_1(\tau)$ can be chosen as

$$P_1(\tau) = (\tau - a)^p (\tau - b)^q \; , \tag{18}$$

where p and q are integers, usually less or equal to 3. The integers p and q depend upon the highest order condition match required at times $\tau = a$ and $\tau = b$ respectively. For example, when boundary conditions at $\tau = 0$ and $\tau = 1$ up to acceleration level must be imposed on the trajectory function, $P_1(\tau)$ can be chosen as $\tau^3 (\tau - 1)^3$. The drawback of this approach is that the after basis functions φ_j are modified by $P_1(\tau)$, it looses the orthogonality, which makes it computationally harder to determine the coefficients for the trajectory solution.

In this paper, we extend the one dimensional function modeling by incorporating the constraints in the basis functions set as explained in the Section IV.A, thus eliminating the need for enforcing constraints by an extra term $P_1(\tau)$.

VI. Aerodynamic Coefficient Function Estimation

An important goal of the ATRC system is the adaptive reshaping of the RLV trajectory in the presence of altered dynamic characteristics of the vehicle when unexpected damage occurs in the various operating scenarios. Any damage to a vehicle that has an impact on the external shape of the vehicle, or that creates an impediment in normal functioning of the control surfaces, results in alteration of the vehicle's aerodynamic characteristics. Figure 7 and Figure 8 show few examples of the Pitch moment coefficient variation in the presence of various damage scenarios. Since the aerodynamic behavior of the vehicle is captured in aerodynamic coefficients that are used for the design of vehicle control and trajectory planning, it becomes mission critical to adapt reference trajectory for the altered vehicle dynamics. Hence we need an approach to build a smooth on-line aerodynamic model. Physics based modeling software such as Missile

DATCOM allows fast computation of aerodynamic coefficients for given flight points and the results can be stored in tabular form. However, for efficient real-time trajectory reshaping application, it is desired to represent aerodynamic coefficients in smooth functional form that are governed by few parameters. In this section, we present a piecewise continuous and smooth function model in two dimensions.

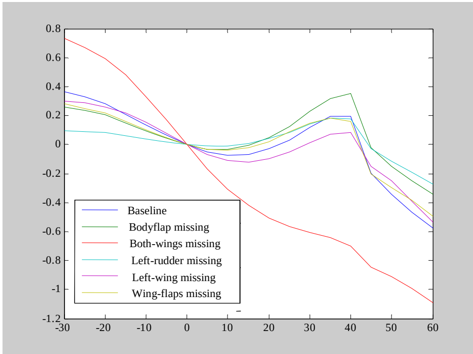


Figure 7. Moment Coefficient with AOA for Nominal and Various Failure Cases

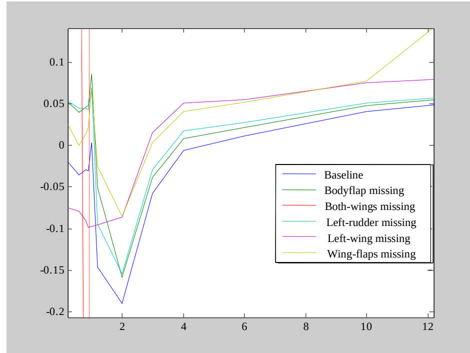


Figure 8. Moment Coefficient with Mach for Nominal and Various Failure Cases

A. Physics Based Aerodynamic Modeling

Given the geometry of the vehicle, very good estimations of the aerodynamic coefficients can be generated based on Physics based modeling in a very small time. For example, if damage on the vehicle is characterized adequately, the DATCOM technology allows specification of the altered geometry of the vehicle to compute the corresponding aerodynamic coefficients at desired points of the flight envelope. In our approach, we generate a large number of data points for the aerodynamic coefficients using DATCOM technology and then fit a piecewise continuous and smooth function to create a functional model for vehicle aerodynamics. The functional form is later used for trajectory reshaping.

B. Finite Element Function Approximation

We formulate the aerodynamic coefficient function with a set of parameters. First we show our approach for piecewise continuous and smooth one-dimensional function, which is later extended to two-dimensions. We used a finite element modeling approach so that the approximation function would capture local variations in an efficient manner. Ref.[10, 11] demonstrates the use of finite element piecewise approximation for mapping geopotential. Ref. [6, 8] applied the technique for aerodynamic coefficients representation. First, the argument space of the function is divided to form a grid with one control point at every grid point. At each control point we use a local polynomial function that is determined using a weighted, least square method from a given set of nominal data generated from DATCOM.

If a function is available at few data points, a continuous function can be obtained by interpolation. Note that interpolation in input space of $\Re^p$ is a surface in $\Re^{p+1}$ space. The interpolation in strict sense can be defined in the following manner. Suppose, given N points: $q_1, q_2, \cdots, q_N \in \Re^p$ and N scalars: $a_1, a_2, \cdots, a_N \in \Re$, we wish to find a function f such that $f(q_i) = a_i$, $i = 1, \cdots, N$. A standard way of interpolation is to use the Lagrange polynomials. However, when number of data points is large, the degree of polynomial becomes high that may result in excessive undesirable modulations in the functions. Another simple approach is to linearly interpolate between the adjacent points. The linear interpolation preserve the function values at the given data points, but smoothness is lost. For smoothness, the continuity of the first derivative of the approximated function must be ensured. In some cases, not only there is more knowledge about the function value, but also we may know or we can estimate the slope of function behavior. For example, the DATCOM model can estimate the aerodynamic coefficients and certain dynamic derivatives at a given flight point. In this case it is desired that besides continuous derivative for smoothness, the values of first derivatives are also preserved. In our functional modeling approach for vehicle aerodynamic model, we wish to preserve the function value and the estimated function slopes.

4. One-Dimensional Finite Element Approximation

A major benefit of using a finite-element approach is that in this approach local approximations for the function model are used that are simpler and more accurate. Here we present a finite element approach for modeling the function. First we define a set of control points that divides the input domain in many finite elements. Next, at each control point we obtain a local polynomial approximation of the function based upon the input data. The scope of each local polynomial centered at the control point lies in between the adjacent grid points (see Figure 9). Notice the overlapping of local polynomials that helps in obtaining a smooth function over the entire range. Once local approximations are determined, a smooth global approximation is obtained as a weighted combination of the local approximations. The smoothness of the approximation function implies that the function, as well as its first derivative, is continuous. Notice that grids are not restricted to be equidistant. To capture nonlinearity effectively, more control points should be placed near high non-linearity. Figure 9 uses a second order polynomial function for local approximations.

A highlight of this Finite Element approximation is that it preserves the local function value and its first derivative at its control point. This is achieved by a smooth weighting function that gracefully goes from unity to zero, from one control point to another, without contributing to a first derivative at both control points. The weighting function for the two overlapping local approximation curves is given as

$$W_1(\overline{q}) = (1-\overline{q})^2(1+2\overline{q}), \quad W_2(\overline{q}) = 1 - W_1(\overline{q}),$$

where $\overline{q}$ is the normalized coordinate with the origin at the *first* control point. Note that the weighting function ensures that $W_1(0)=1$, $W_2(0)=0$ at the first control point and similarly, $W_1(1)=0$, $W_2(1)=1$ at the second control point. If $f_1(\overline{q})$ and $f_2(\overline{q})$ are two local approximations, the weighted global approximation $F(\overline{q})$ between two control points is given as

$$F(\overline{q}) = W_1(\overline{q})f_1(\overline{q}) + W_2(\overline{q})f_2(\overline{q}).$$

Figure 10 shows the final approximated function that is made of the weighted combinations of the local function approximations.

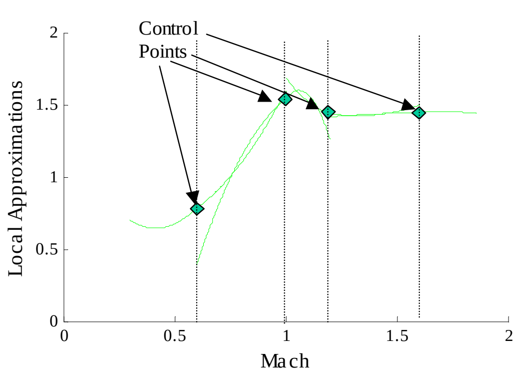


Figure 9: Local Approximations Centered at Control Points

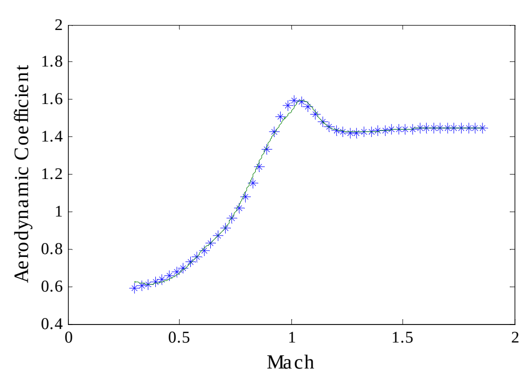


Figure 10: Smooth Functional Approximation using Weighted Local Approximations

5. Multi-Dimensional Finite Element Approximation

In this section we extend the formulation for FEM modeling for multi dimensional functions. Let F be given as

$$F = F(q_1, q_2, \cdots, q_n) \quad (19)$$

and we have to determine an estimate $\hat{F}(q_1, q_2, \cdots, q_n)$ from a finite element model. As in one-dimensional case, we assume that the domain of F is covered with finite number of equidistant nodes. Each preliminary local approximation $F_{i_1,i_2,\cdots,i_n}$ is valid in the $2\times 2\times\cdots\times 2$ hypercube centered at a node represented by n indices given as $(i_1, i_2, \cdots, i_n)$. Each index is numbered in increasing order for the nodes lying in its dimension. Consider a hyper cube formed by 2^n nodes, each node representing a corner of the hyper cube. The final approximation $F_{i_1,i_2,\cdots,i_n}$ is valid in the unit hypercube whose "lower left corner" is $(i_1, i_2, \cdots, i_n)$. We shift the origin of the coordinates to the node $(i_1, i_2, \cdots, i_n)$ that has the lowest indices in each dimension. The local coordinates are normalized as

$$\overline{q}_j = \frac{q_j - q_{j,i_j}}{q_{j,i_j+1} - q_{j,i_j}} . \quad (20)$$

Let us define a function W as a function of the normalized coordinate $\overline{q}_j$ as

$$W(\overline{q}_j) = (1-\overline{q}_j)^2(1+2\overline{q}_j) . \quad (21)$$

Now the weight function for an arbitrary corner node $(i_1, i_2, \cdots, i_k+1, i_{k+1}+1, \cdots, i_n)$ of the hyper cube can be written as

$$w_{i_1,i_2,\cdots,i_k+1,i_{k+1}+1,\cdots,i_n} = W(\overline{q}_1)W(\overline{q}_2)\cdots W(1-\overline{q}_k)W(1-\overline{q}_{k+1})\cdots W(\overline{q}_n) ,$$
$$\text{for} \quad 0 \le \overline{q}_j \le 1, \qquad j = 1\cdots n . \quad (22)$$

The weight function defined by Eq. (22) has the following properties:

- Weight is unity at its own node and, it is zero at any other node or edge formed by other nodes.
- Summation of all the weights corresponding to the corner nodes of the hyper cube at any point inside the hyper cube is equal to one.

The estimated function $\hat{F}$ can be defined in terms of normalized coordinates centered at node $(i_1, i_2, \cdots, i_n)$ as

$$\hat{F}(\overline{q}_1, \overline{q}_2, \cdots, \overline{q}_n) = \sum_{j=1}^{2^n} w_{i_1,i_2,\cdots,i_n,j} F_{i_1,i_2,\cdots,i_n,j} . \quad (23)$$

6. An Example of Finite Element Function Approximation for Functions of Two Variables.

Let *Z* be the matrix of values of a function sampled at the grid points of a 2D space defined by *x* and *y*. We are interested in representing the function *Z* in closed form, using 2D polynomial approximation. The closed form representation can be used to evaluate the function *Z* at arbitrary *x* and *y* points. We approximate the function *Z* using a number of 2D polynomial function elements of a smaller area of support. The number of function elements to be used and the degrees of the polynomials in *x* and *y* are inputs to the algorithm. We specify this by the *x* and *y* coordinates of the control points as separate vectors. Note that the control points lie exactly on the grid points. At these data points, the approximated value of the function *Z* equals the value of the local function centered at that control point. The chosen approximation is locally optimal in the least squared error criteria (i.e., within the area of support of each function element).

To formulate finite element function approximation satisfying certain constraints, we proceed as follows. The orthonormal basis set of functions are computed that satisfy the constraints and then searched for local functions that are optimal in the vector space spanned by the constrained orthonormal basis functions, based on least square optimality criteria. More details on the finite element function approximation can be found in [7].

Concentrating on the finite element function approximation method for 2D functions, we initially proceeded by calculating the weights in the x and y directions separately. The actual weight function at each point in the grid is the product of the x and y weights at that location. The x and y weight functions satisfy the criteria that at the current control point location, it has a value of one and gradually decreases to zero as it moves away. The products of the 1D weight functions also satisfy the same criteria in the 2D x-y plane. The intended consequence of this property is that the approximated function value will be equal to the value of the local function at the control point.

Figure 11 shows an example of a weight function in the 2D plane that is the product of the two 1D weight functions.

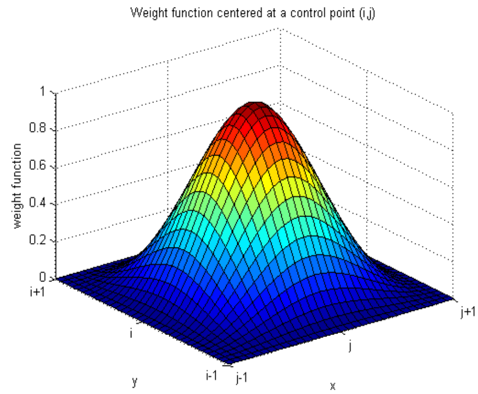


Figure 11. Weight function centered at a control point. The value is one at the location of the control point and gradually decreases and reaches zero at the neighboring control points.

Figure 12 shows the results of finite element approximation for the CM coefficient for a non-damaged (baseline) spacecraft with 16 control points along the Mach axis and 21 control points along the alpha axis. The basis functions were polynomials of degree 2 in x and 2 in y. .

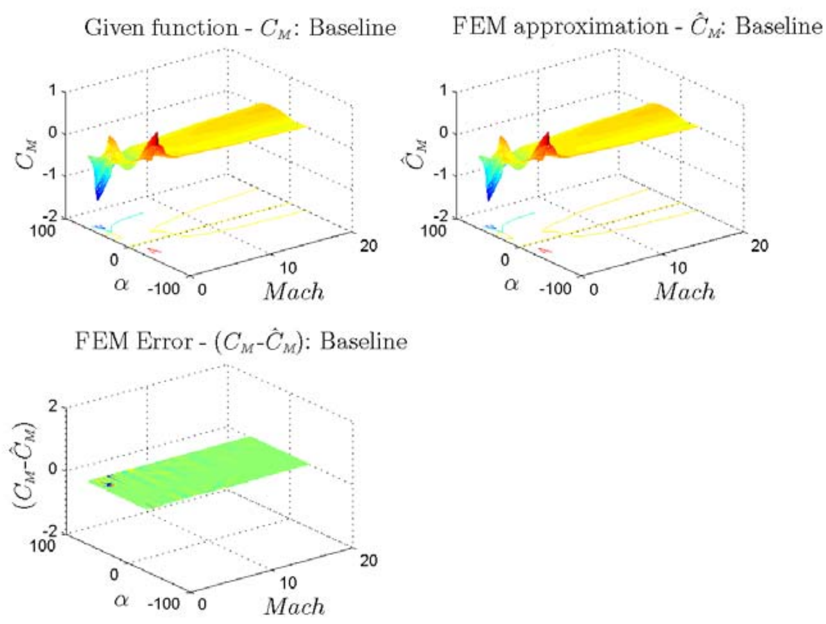


Figure 12. Finite element function approximation. (a) Given function the CM coefficient for baseline (b) FEM Approximated Function (c) Error between the given and approximated functions. 16 control points along the Mach axis and 21 control points along the alpha axis were used for FEM approximation. Each local function is a polynomial of degree 2 in x and 2 in y.

VII. Conclusion

In this paper we addressed the function modeling techniques to facilitate an efficient on-line inverse dynamics methodology for trajectory reshaping of the RLVs. We presented a technique for generating orthonormal polynomial basis functions with in-built function constraints at divers points of the function. The general approach allows the constraints ranging from the function value, to constraints on values of the higher derivatives of the functions. Next we presented a finite element approach for modeling multidimensional, piecewise continuous and smooth functions The use of FEM based smooth modeling and use of constrained orthonormal basis function are expected to improve the convergence of iterative numerical solution for dynamic inversion problem.

Acknowledgments

The research was supported by the AFRL/VACA , WPAFB, Ohio, under contract number FA8650-04-M-3428

References

[1] Gao, C., and Hess, R.A., "Inverse Simulation of Large-Amplitude Aircraft Maneuvers," *Journal of Guidance, Control, and Dynamics*, Vol. 16, No. 4, 1993, pp. 733-737.

[2] Verma A., "Determination and Stable Tracking Of Feasible Aircraft Trajectories Using Adaptive Inverse Dynamics", Ph.D. Dissertation, Texas A&M University, May 2000.

[3] Junkins, J.L. and Kim, Y., "Stability and Control of Robotic Space Manipulators", *Teleoperation and Robotics in Space*, edited by S. B. Skaar and C. F. Ruoff, Vol.161, Progress in Astronautics and Aeronautics, AIAA, Washington DC, 1995, pp.315-350.

[4] Fliess, M., Lévine, J., Rouchon, P., "Generalized State Variable Representation for a Simplified Crane Description", *International Journal of Control*, Vol. 58, No.2, 1993, pp. 277-283.

[5] Nieuwstadt, M.V., Rathinam, M., Murray, R.M., "Differential Flatness and Absolute Equivalence", *Proc. IEEE Control and Decision Conference*, Lake Buena Vista, FL, Dec. 1994, pp.326-333.

[6] Nieuwstadt, M.V., Murray, R.M., "Real Time Trajectory Generation for Differentially Flat Systems", *International Journal of Robust Nonlinear Control*, Vol.8, Issue 11, September 1998, pp.995-1020.

[7] Bortoff, S.A., "Path Planning for UAVs Using Potential Fields" 2000 American Control Conference, Chicago, IL, June 28-30, 2000

[8] Blake, William B., "Missile DatCom User's Manual – 1997 Fortran 90 Revision," Final Report, 1998.

[9] Oppenheimer, M.W., Doman, D.B. and Bolender, M.A., "A Method for Estimating Control Failure Effects for Aerodynamic Vehicle Trajectory Retargeting," 2004, AIAA Paper 2004-

[10] Jancaitis, J.R., and Junkins J.L., "Modeling N Dimensional Surfaces Using a Weighting Function Approach," Journal of Geophysical Research, Vol. 79, August 1974, pp. 3361.

[11] Junkins J.L., "Investigation of Finite-Element Representations of the Geopotential," AIAA Journal, Vol. 14, No. 6, June 1976, pp. 803-808.